F**UNCTII PENTRU OPERATII CU SIRURI DE CARACTERE**

Functiile pentru operatii cu siruri se gasesc in header-ul **<string.h>.**   
  
**ØFunctia strlen**   
int strlen(nume\_sir);– returneaza lungimea efectiva a unui sir (fara a numara terminatorul de sir).  
*Exemplu*:  
char a[50]=”ora de informatica”;àstrlen(a) = 18   
  
**ØFunctia strcpy**   
strcpy(sir\_destinatie,sir\_sursa); – copiaza sirul sir\_ sursain sir\_destinatie (se simuleaza atribuirea a=b).  
  
ATENTIE!! Nu este permisa atribuirea intre doua siruri de caractere folosind operatorul =. Atribuirea se face folosind functia strcpy.  
  
*Exemplu:*  
char a[50]=”primul sir”,b[40]=”al doilea sir”;  
a=b; *eroare*  
strcpy(a,b); à a = ”al doilea sir”; b=”al doilea sir”;   
  
**ØFunctia strcat**   
strcat(dest,sursa);– adauga sirului dest sirul sursa.Sirul sursa *ramane nemodificat. Operatia se numeste* concatenare *si nu este comutativa.*  
Exemplu:  
char \*a=”vine ”,\*b=”vacanta?”;  
strcat(a,b); à a = ”vine vacanta?”;   
  
**ØFunctia strncat**   
strncat(dest,sursa,nr);– adauga dest primele nr caractere din sirul sursa.Sirul sursa ramane nemodificat.   
Exemplu:  
char \*a=”vine ”,\*b=”vacanta?”;  
strncat(a,b,4); à a = ”vine vaca”;   
  
**ØFunctia strchr**   
strchr(sir,c);*– are rolul de a cauta caracterul c in sirul sir. Cautarea se face de la stanga la dreapta, iar functia intoarce adresa subsirului care incepe cu prima aparitie a caracterului c. Daca nu este gasit caracterul, functia returneaza 0. Diferenta dintre adresa sirului initial si cea a subsirului returnat reprezinta chiar pozitia caracterului cautat in sirul dat.*  
Exemplu*:*  
char \*a=”acesta este un sir”,b=’t’,c=’x’,d;  
cout<<strchr(a,b); à se tipareste ”ta este un sir”;  
cout<<strchr(a,c); à nu se tipareste nimic (se tipareste 0 daca se face o conversie la int a lui strchr(a,c) ;  
d= strchr(a,b);  
cout<<”Caracterul apare prima data la pozitia ”<<d-a;

Ex: Sa se afiseze toate pozitiile unui caracter intr-un sir   
#include <iostream.h>  
#include <string.h>  
void main()  
{char a[100],\*p,c;  
cin.get(a,100);  
cin>>c;  
p=strchr(a,c);  
while (p)  
{cout<<"Pozitia "<<p-a<<endl;  
p++;  
p=strchr(p,c);}}  
  
**ØFunctia strrchr**   
strrchr(sir,c);– are acelasi rol cu strchr, cu deosebirea ca returneaza adresa ultimei aparitii a caracterului (cautarea se face de la dreapta spre stanga; r = right)   
  
**ØFunctia strcmp**   
int strcmp(sir1,sir2);– are rolul de a compara doua siruri de caractere. Valoarea returnata este <0 (daca sir1<sir2), =0 (daca sir1=sir2) si >0 (daca sir1>sir2). Functia strcmpface distinctie *intre literele mari si cele mici ale alfabetului.*  
*Obs: Functia strcmp returneaza diferenta dintre codurile ASCII ale primelor caractere care nu coincid*  
  
**ØFunctia stricmp**   
int stricmp(sir1,sir2);– are acelasi rol cu strcmp*, cu deosebirea ca* nu face distinctie *intre literele mari si cele mici ale alfabetului (i = ignore).*  
  
**ØFunctia strstr**   
strstr(sir1,sir2);– are rolul de a identifica daca sirul sir2este subsir al sirului sir1. Daca este, functia returneaza adresa de inceput a subsirului sir2in sirul sir1, altfel returneaza adresa 0. In cazul in care sir2apare de mai multe ori in sir1, se returneaza adresa de inceput a primei aparitii. Cautarea se face de la stanga la dreapta   
  
**ØFunctia strtok**   
strtok(sir1,sir2);– are rolul de a separa sirul sir1in mai multe siruri (cuvinte) separate intre ele prin unul sau mai multe caractere cu rol de separator. Sirul sir2este alcatuit din unul sau mai multe caractere cu rol de separator.  
Functia strtok actioneaza in felul urmator:   
oPrimul apel trebuie sa fie de forma strtok(sir1,sir2); Functia intoarce adresa primului caracter al primei entitati. Dupa prima entitate, separatorul este inlocuit automat prin caracterul nul.   
oUrmatoarele apeluri sunt de forma strtok(NULL,sir2); De fiecare data, functia intoarce adresa de inceput a urmatoarei entitati, adaugand automat dupa ea caracterul nul.   
oCand sirul nu mai contine entitati, functia returneaza adresa nula.

Exemplu*:*  
Sa se separe cuvintele dintr-un text.  
Sa se separe cuvintele dintr-un text.  
#include <iostream.h>  
#include <conio.h>  
#include <string.h>  
void main()  
{char text[100],cuv[10][10],\*p,\*r,separator[]=",. !?";int i=0,nr=0;  
clrscr();  
cout<<"Dati sirul:";cin.get(text,100);  
strcpy(p,text);  
p=strtok(p,separator);  
while (p)  
{strcpy(cuv[++nr],p);  
p=strtok(NULL,separator);}  
cout<<"Sunt "<<nr<<" cuvinte:"<<endl;  
for (i=1;i<=nr;i++) cout<<cuv[i]<<endl;  
getch();}  
  
**ØFunctia strspn cu forma generala**  
int strspn(sir1,sir2);– are rolul de a returna numarul de caractere ale sirului sir1(caractere consecutive care incep obligatoriu cu primul caracter) care se gasesc in sirul sir2.   
*Exemplu*:  
strspn(“AB2def”,”1B3AQW”); à returneaza 2, pentru ca primele 2 caractere ‘A’ si ‘B’  
din sir1 se gasesc in sir2.  
strspn(“FAB2def”,”16A32BF”); à returneaza 0, deoarece caracterul ‘F’ cu care incepe sir1nu se gaseste in sir2.   
  
**ØFunctia strcspn cu forma generala**  
int strspn(sir1,sir2);– are rolul de a returna numarul de caractere ale sirului sir1(caractere consecutive care incep obligatoriu cu primul caracter) care *nu* se gasesc in sirul sir2.   
*Exemplu*:  
strspn(“AB2def”,”123”); à returneaza 2, pentru ca primele 2 caractere din sir1 nu se gasesc in sir2.  
//Se citeste un sir de caractere care nu contine caractere albe. Sa se decida daca sirul este alcatuit exclusiv din caractere numerice.  
#include <iostream.h>  
#include <conio.h>  
#include <string.h>  
void main()  
{char text[100],cifre[]="0123456789";  
clrscr();  
cout<<"Dati sirul:";cin.get(text,100);  
if (strcspn(cifre,text)==strlen(text))  
cout<<"exclusiv numeric";  
else cout<<”nenumeric”;  
getch();}  
  
  
  
**ØFunctia strlwr** cu forma generala  
strlwr(sir);– are rolul de a converti toate literele mari din sirin litere mici. Restul caracterelor raman neschimbate.  
**ØFunctia strupr** cu forma generala  
strupr(sir);– are rolul de a converti toate literele mici din sirin litere mari. Restul caracterelor raman neschimbate  
**ØFunctia strbrk** cu forma generala  
strpbrk(sir1,sir2);– actioneaza in felul urmator:   
oCauta primul caracter al sirului sir1in sir2. Daca este gasit, returneaza adresa sa din cadrul sirului sir1si executia se termina. Altfel, se trece la pasul urmator.   
oCauta al doilea caracter al sirului sir1in sir2. Daca este gasit, returneaza adresa sa din cadrul sirului sir1si executia se termina. Altfel, se trece la pasul urmator.   
o…   
oDaca nici un caracter al sirului sir1nu apartine sirului sir2, functia returneaza adresa nula.  
**ØFunctia atof** cu forma generala  
double atof(sir);– converteste un sir catre tipul double. Daca aceasta conversie esueaza (se intalneste un caracter nenumeric), valoarea intoarsa este 0. Aceasta functie (ca si cele similare) necesita includerea librariei stdlib.h.  
  
**ØFunctia \_atold** cu forma generala  
long double \_atold(sir);– converteste un sir catre tipul long double. Daca aceasta conversie esueaza, valoarea intoarsa este 0.  
  
**ØFunctia atoi** cu forma generala  
int atoi(sir);– converteste un sir catre tipul int. Daca aceasta conversie esueaza (se intalneste un caracter nenumeric), valoarea intoarsa este 0.  
  
**ØFunctia atol** cu forma generala  
long atol(sir);– converteste un sir catre tipul long. Daca aceasta conversie esueaza (se intalneste un caracter nenumeric), valoarea intoarsa este 0.  
  
**ØFunctia itoa** cu forma generala  
itoa(int valoare,sir,int baza);– converteste o valoare de tip int in sir, care este memorat in variabila sir. Baza retine baza de numeratie catre care sa se faca conversia. In cazul bazei 10, sirul retine si eventualul semn -.  
  
**ØFunctia ltoa** cu forma generala  
ltoa(long valoare,sir,int baza);– converteste o valoare de tip long intin sir, care este memorat in variabila sir.  
**ØFunctia ultoa** cu forma generala  
ultoa(unsigned long valoare,sir,int baza);– converteste o valoare de tip unsigned long in sir, care este memorat in variabila sir.